CIS 530 Final Project: Ingredient Replacement

Leon Zhou
zhliyang@seas.upenn.edu

Anshul Wadhawan
anshulw@seas.upenn.edu

Anni Pan
annipan@seas.upenn.edu

Abstract

We present several models for the ingredient replacement module in Taskbot that recommends substitutes for ingredients that the user does not have. We started with a simple co-occurrence matrix based model, then moved on to other unsupervised learning models such as GPT-3 prompting, KNN with FoodBERT embeddings, and finally tried a supervised learning model: contextualized ingredient replacement. Among these models, the GPT-3 prompting model achieved the highest accuracy of 75.5%.

1 Introduction

While Taskbot can walk us through popular recipes step-by-step, we don't live in a perfect world where we have all the ingredients lying around to complete an exotic recipe. Constraints are commonplace when trying to follow a recipe, and it's a very frequent scenario is that some of the ingredients are missing. Our goal is to develop an ingredient replacement module for Taskbot that recommends substitutions for ingredients that the user does not have. An example usage of the module would be:

Bot: Season the chicken with $\frac{1}{2}$ teaspoon of kosher salt per 1 pound.

User: I don't have kosher salt.

Bot: You can replace it with table salt or sea salt.

As we need to extract the ingredient name from the user input, train a model based on recipe data, and output the most suitable substitutions, this is a natural language processing task.

Specifically, extracting the ingredient name can be seen as a special case of named entity recognition that aims to tag the appropriate words in the sentences as ingredient names. Outputting a list of substitutions can be seen as finding the closest ingredients to an ingredient based on their meaning and usage, which is also an important and common task in the NLP world.

To simplify the problem a bit, our goal is that, given an input ingredient name, output a list of substitutions for that ingredient. For example, in the previous example, the input would be `"kosher salt"`, and the output should be `["table salt", "sea salt"]`.

The reason for picking this task is that, as we mentioned before, missing ingredients is a common scenario when cooking, and it would make the Taskbot more usable if the user can simply interact with the Taskbot to figure out the replacement for the missing ingredient instead of doing another search on the Internet. It is also a functionality that can extend to other use cases, for example: finding substitutions for milk when the user is lactose intolerant, and recommending cooking utensil replacements.

2 Literature Review

2.1 Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images (Marin et al., 2019)

The paper trains a neural network to learn a joint embedding of recipes and images for the image-recipe retrieval task: namely, retrieve the recipe from the food image. The recipe model includes two decoders: one for ingredients and another for instructions. The ingredients encoder combines the sequence of ingredient word vectors using a bidirectional LSTM model, since the ingredient list is an unordered set. The instructions encoder is a forward LSTM model over instruction vectors. The outputs of both encoders are concatenated and embedded into a recipe-image joint space, where the image representation is projected into through a linear transformation.

The measures used in the experiments are median rank (MedR) and recall rate at top K (R@K),

where K = 1, 5 and 10. The joint neural embedding model performed better than the baseline model (Canonical Correlation Analysis) in all measures, and adding semantic regularization to the model resulted in even greater improvement. In 65% of all evaluated queries, it retrieved the correct recipe given a food image. With semantic regularization, the model surpasses humans' performance by 6.8 percentage points.

2.2 Exploiting Food Embeddings for Ingredient Substitution (Chantal Pellegrini and Groh, 2021)

This paper presents two models for ingredient embeddings, Food2Vec and FoodBERT. They train two models, word2vec and BERT, on recipe instructions from the Recipe1M+ dataset to compute meaningful ingredient embeddings. Additionally, they combine these text-only approaches with an image-based approach resulting in multimodal ingredient representations. They also use FoodBERT to perform relation extraction on recipe comment data to extract substitute pairs. They present the following approaches that can be used for context-free ingredient substitute generation:

Pattern Extraction. One previously applied method for food substitute extraction, which they use as a baseline, is to use user comments from recipe sites since a good portion of them mention how to replace certain ingredients with others. They use `Spacy` to extract these recommendations by applying the hand-crafted patterns.

Food2Vec. The Food2Vec approach is separated into two parts: The first part calculates text-based embeddings for all ingredients and optionally concatenates them with image based embeddings. The second part uses these embeddings in conjunction with KNN to predict substitutes. For generating substitutes, they search for the N nearest neighbors in the embedding space containing all ingredients. N functions as a threshold to balance the number of proposed substitutes and the model's precision.

FoodBert. The FoodBERT approach is separated into two parts: The first part calculates text-based embeddings for up to 100 occurrences of every ingredient and optionally concatenates them with image-based embeddings. The second part employs these embeddings together with KNN and a further scoring and filtering step to predict substitutes.

Multimodel Representations. They concatenate the text-based and image-based ingredient representations to get one multimodal embedding of size 200 for Food2Vec or 1,536 for FoodBERT.

2.3 Enabling Language Models to Fill in the Blank (Chris Donahue, 2020)

This paper introduces a LM-agnostic framework to leverage language models to complete the "fill in the blank" task. Previously in Food-bert, we see that ingredient substitutions are performed context-less by finding the closest ingredient substitutes in the word embedding space. While this is an effective approach to gauge relationships between different ingredients, not taking into consideration the role of a specific ingredient in a recipe has an adverse effect on providing a decent ingredient replacement suggestion. Following the methodology introduced in the paper, we formulate the ingredient substitution task as a fill-in-the-blank task to utilize word embeddings in its natural setting (inside LMs) and get context information as part of the input.

3 Experimental Design

3.1 Data

3.1.1 Data for Unsupervised Learning models

For the unsupervised learning models, we are using the pre-processed dataset *simplified-recipes-1M* (Schmidt, 2019), which cleans the raw recipes in *recipes 1M dataset* (Marin et al., 2019) and extracts the most frequent ingredients. It contains an `ingredients` array that stores the names of all the ingredients that occur in the dataset, and a `recipe` array in which each entry is a list of integers, each representing the ingredient's index in the `ingredients` array.
A sample from the `ingredients` array: `["salt", "pepper", "butter", ...]`. A sample from the `recipe` array: `[208, 473, 347, 55, 63, 173, 3, ...]`. Some statistics for the `recipe` array can be found in the following table:

Statistics	Value
Number of recipes	1067557
Avg number of ingredients per recipe	16.5

To evaluate our model on test data, we need to curate a dataset that contains the ground truth labels. We are using the ground truth dataset

collected by Pellegrini et al., 2021, cleaned so that it only contains the ingredients present in our `ingredients` array. To create the ground-truth sets, they selected 42 ingredients, looked up their substitutions in the Food Substitutions Bible (Joachim, 2010), and continuously extended the dataset by manually adding any correct predictions of their models if they were missing in the ground truth. The final ground truth dataset contains 38 ingredients, where each ingredient is mapped to a list of its substitutions. An example from the test set is: `"american cheese": ["cheddar cheese", "colby cheese", "muenster cheese", ...]`

3.1.2 Data for Supervised Learning models

For our experiment with the supervised fill-in-the-blanks models (section 4.2.3), we also need to curate a labeled dataset where the data are recipes with their ingredients masked and training objective is to uncover these masked ingredients. To get the instructions containing the ingredients, we used one component of *Recipes 1M*: `recipes_with_nutritional_info.json`, which contains an array of recipes with information such as the ingredients, ingredient quantities, instructions and nutrition per ingredient.

As we are only interested in the instructions, we extracted the instructions of from each recipe one line at a time and split these recipes instructions into the training, validation and test sets in a ratio of 80:10:10. In order to conform to the data format specified by Chris Donahue (2020), we outputted these recipe instructions into three files, `train.txt`, `valid.txt` and `text.txt`, with three newline characters separating each recipe. The number of recipes for the three files are 40988, 5123 and 5124, respectively. However, due to limits on computational power, we had to train our model with only 1000 training samples and 100 validation samples.

With the unlabeled raw data ready, we then created a module for ingredient detection trained on the `det_ingrs.json` of recipe1M+, where each example (short noun phrase) in the training example is assigned a binary label denoting whether its referring to a food ingredient. We trained a bi-LSTM model on the ingredient detection task and achieved 99.8 on the validation set. In order to mask all ingredients in a given recipe, we first run a nltk pipeline to obtain POS tags of all the words, from there we parse the POS tagged sentence to get noun phrases. Our ingredient detection module will then determine if a given noun phrase refers to an ingredient. Following the custom mask function in ILM codebase Chris Donahue (2020), we were able to mask most of the ingredients in our recipe corpus.

3.2 Evaluation Metric

Our evaluation metrics include the over all precision and recall on the test set, and precision and recall for each ingredient in the test set.

Let the set G be the set of all correct substitution pairs as specified in the ground truth dataset. Let the set P be the set of all predicted substitution pairs outputted by our model. The overall precision is calculated by $\frac{|P \cap G|}{|P|}$, and the overall recall is calculated by $\frac{|P \cap G|}{|G|}$.

The per-ingredient recall and precision are defined similarly as above, with the scope limited to each individual ingredient: it calculates the precision and recall for each ingredient at a time and outputs a dictionary that maps each ingredient to the precision and recall of its predicted substitutes.

3.3 Simple Baseline

To construct a simple baseline, we are using the co-occurrence matrix to model how often any two ingredients co-occur, normalized the matrix, and predict an ingredient's substitutes to be the five other ingredients that co-occur most frequently with the ingredient. Some outputs of the model is recoreded below:

Ingredient	Substitutes
Duck	Duck sauce, goose, confit
Hazelnut	Ground hazelnuts, hazelnut liqueur
Salmon	Smoked salmon, canned salmon
American cheese	luncheon meat, bologna

The model successfully captures some of the substitution information, such as `chicken` is a substitution for `chicken gravy` and `cheese` is a substitution for `jack cheese`. However, some results that the model outputs are actually not substitutions: for example, `luncheon meat`, `bologna` are outputted for `American cheese`, which should be better interpreted as that the cheese are seasonings for luncheon meat

and bologna instead of substitutions for bacon. We do acknowledge that this baseline is biased since ingredients that co-occur with each other are not necessarily substitutes for each other.

For the baseline model, the overall precision is 0.1 and the overall recall is 0.029, indicating the difficulty of getting a high precision and recall for this task: each ingredient has more than ten substitutions in the ground truth dataset on average. The flaws in the baseline model also points out ways for further improvement: context-based models should be a more accurate way of measuring similarity between ingredients.

4 Experimental Results

4.1 Published Baseline: GPT3 Prompting with Single Answer

The strong baseline we used is GPT3 prompting. Specifically, we treated the ingredient replacement task as a text completion problem using the "davinci" engine. Under this few-shot learning setting, where the model are presented as few as 7 examples (carefully selected to not overlap with the test set), the large language model was able to perform at a relatively high precision: 0.894, albeit at a low recall:0.052. That is because the gpt3 model is prompted to return only a single ingredient replacement:
Q:What is a good replacement for ingredient1?
A: ingredient2.
Consequently, the recall rate can be much higher once we compile a better set of few shot examples with multiple ground truth replacements.

4.2 Extensions

4.2.1 K-Nearest Neighbors Model

A subsection of the FoodBERT embedding space, after reducing the embedding dimensions from 78600 to 3 using t-SNE (Cai and Ma, 2021), has been showcased in Figure 1. An interactive analysis of the space has been presented at the link[1]. The analysis shows different clusters like {`onion`, `green onion`, `white onion`}, {`oil`, `cooking oil`}, {`salt`, `seasoning`, `garlic salt`}, and {`cream`, `heavy cream`, `ice cream`}, each containing closely placed ingredients. This shows that similar ingredients tend to occupy similar spots in the embedding space even when dimensionality is reduced significantly.

[1] https://tinyurl.com/4kfbmadt

The proposed model finds K nearest neighbors for a particular ingredient in the FoodBERT embedding space, and shortlists them on the basis of a thresholding criteria.
The final approach is separated into two parts:

- The first part calculates text-based embeddings for up to 100 occurrences of every ingredient and optionally concatenates them with image-based embeddings.
- The second part employs these embeddings together with KNN and a further scoring and filtering step to predict substitutes.

The model is able to generate decent substitutes for common ingredients. Some examples of generated substitutes are given in Table 2.

Ingredient	Substitutes
Salt	Seasoning, Pepper
Sugar	Sweetener, Splenda
Honey	Maple Syrup, Marmalade, Syrup
Pepperoni	Salami, Pepperoncini

For this model, the overall precision is 0.784 and the overall recall is 0.139. There are significant improvements in terms of precision, however, similar pattern is not observed in recall. This is again due to the fact that the ground truth dataset has a high number of substitutions laid out for each ingredient, while the model only predicts a small number of them.

4.2.2 GPT3 Prompting with Multiple Answers

To account for the low recall issue with our previous gpt3 prompting design. We changed the prompt design to return multiple ingredient replacements:

Q: What are some good replacements for cheddar cheese?A: yellow cheddar, colby cheese, double gloucester cheese, brick cheese, tillamook cheese.
Q: What are some good replacements for strawberries?A: raspberry, kiwi, rhubarb, figs.
Q: What are some good replacements for chicken?A: ground turkey, beef, pork, lamb.
.....
The new gpt3 prompting method tripled the recall at the cost of 10 pnts of precision. The large fluctuation warrants the need for us to construct a

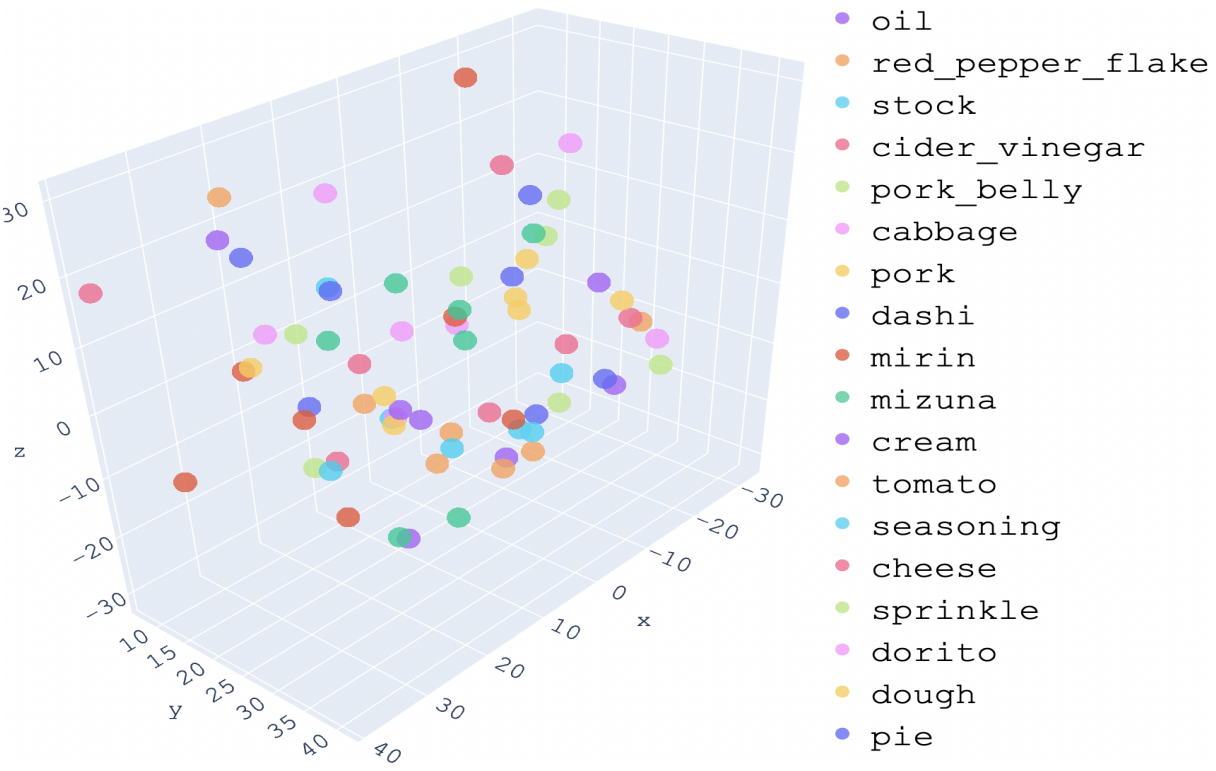

Figure 1: FoodBERT Embedding Space

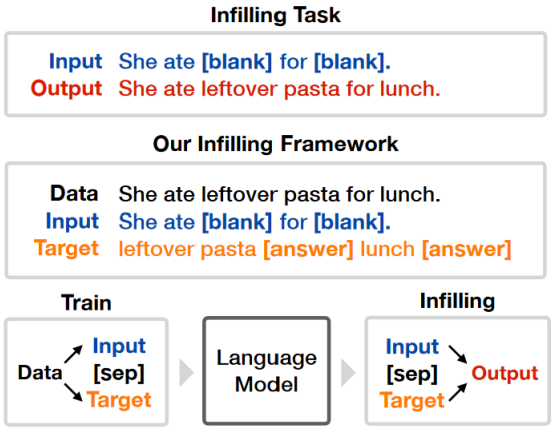

Figure 2: Infilling by Language Modeling

larger test set. We were able to achieve precision = 0.755, recall = 0.161 with the new prompt design.

4.2.3 Contextualized Ingredient Replacement

Our previous approaches as well as all existing methods on ingredient replacement does not take into account the context in which the ingredient appears in. e.g. When the user asks for a substitution for chicken in a meaty recipe, tofu may not be a good ingredient replacement.

To give contextualized ingredient replacements, we decided to leverage large language models in our task through a novel "fill-in-the-blank" problem formulation (see figure 2) (Chris Donahue, 2020), where inside each recipe, the ingredients are detected and masked. Then, the language model attempt to recover the masked ingredients as its training objective (Infilling by Language Modeling). During inference, the user's query will be masked and the predictions for the masked ingredients will serve as the model's replacement suggestions.

We resumed training from the author's(Chris Donahue, 2020) ILM model based on gpt2, and was able to obtain reasonable ingredient replacements (see table 1). However, so far we were only able to mask 1000 recipes for training. In future work, we will finish labelling all 40k or so recipes before resume training with gpt2.

Stir the salsa, soup, sour $\langle masked \rangle$, cheese, corn, $\langle masked \rangle$ and beans in a large bowl.	masked input
$\langle SEP \rangle$ cream$\langle SEP \rangle$ crust$\langle SEP \rangle$	prediction
Cream the cornmeal,$\langle masked \rangle$, and $\langle masked \rangle$ together.	masked input
$\langle SEP \rangle$ salt$\langle SEP \rangle$ butter$\langle SEP \rangle$	prediction

Table 1: ILM model on recipe1M+.

4.3 Error Analysis

The best-performing model, GPT-3 prompting, achieves 75.5% accuracy and 16.1% recall. A sample of the predicted ingredients that are not in the ground truth dataset can be found below:

Ingredient	Substitutes
cheddar	colby cheese, brick cheese, gouda cheese
kale	collard greens, swiss chard, mustard greens
raisins	dried cherries, dried blueberries
strawberry	figs, rhubarb, kiwi

Though these ingredient substitutions are not in the ground truth dataset, they are actually good substitutes for the given ingredients. This observation suggests that the ground truth dataset needs to be further expanded, and we should use some human evaluations to evaluate the models. Therefore, the GPT-3 model does a better job of recommending substitutes than its actual accuracy on the ground truth dataset suggests.

On the other hand, a sample of the ingredients that are in the ground truth dataset but missing in the predictions can be found below:

Ingredient	Substitutes
American cheese	monterey jack, provolone
chicken	rabbit, capon, shrimp
peanut	pistachio, macadamia
raspberry	huckelberry, boysenberry

A majority of these substitutions are not very commonplace ingredients, and therefore it does not hurt the model's performance in real-life situations even when it fails to predict these ingredients. Moreover, since the ground truth data has an average of more than 10 substitutes for each ingredient, getting a high recall is both hard and unnecessary: we do not need to provide so many substitution choices for the user. Therefore, though the GPT-3 prompting model's performance metrics are not excellent on the test set, it will work reasonably well in real-life scenarios.

For our fill-in-the-blank model, while it ideally will take into account the context of the ingredient replacement and give better recommendations, our current training set is very limited (it would take another 10 hours to annotate/mask ingredients) and we are unable to showcase the full-power of this problem formulation. Moreover, our current evaluation script is context-less and doesn't work with our ILM model. We also noticed in many of the recipe instructions, there are simply not enough hints/context even for a human to guess the masked ingredient, in future work, we will include additional signals such as ingredients used in the recipe and recipe name. Lastly, foodbert is a more suitable language model than gpt2 for our use case.

5 Conclusions

In this paper, we propose different methodologies to tackle the task of finding substitutes for ingredients required in recipes, as and when asked by the user while interacting with voice based artificial agents. We propose several models, ranging from a simple co-occurrence matrix based model, GPT-3 prompting, K-Nearest Neighbors algorithm on FoodBERT embeddings, and context based ingredient replacement. The GPT-3 prompting model achieving an accuracy of 75.5% was the best among all the proposed methodologies.

References

T. Tony Cai and Rong Ma. 2021. Theoretical foundations of t-sne for visualizing high-dimensional clustered data.

Monika Wintergerst Chantal Pellegrini, Ege Ozsoy and Georg Groh. 2021. Exploiting food embeddings for ingredient substitution. *In Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies - HEALTHINF, 67-77*.

Percy Liang Chris Donahue, Mina Lee. 2020. Enabling language models to fill in the blanks. *arXiv:2005.05339*.

David Joachim. 2010. *The Food Substitutions Bible: More Than 6,500 Substitutions for Ingredients, Equipment and Techniques*.

Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Trans. Pattern Anal. Mach. Intell.*

Dominik Schmidt. 2019. simplified-recipes-1m dataset.

Acknowledgments

We would like to thank Professor Mark Yatskar for their continued guidance towards the completion of this project.